Docker was used to containerize the application and encapsulate the environment in order to construct container images for the Personal Finance Management System (PFMS) deployment on AWS Elastic Beanstalk:

Steps:

1. Dockerfile Creation: To specify the actions required to construct the application's container image, a Dockerfile was created. The application's JAR file is copied into the container, starting with a base Java image, and the command to begin the application within the container is specified.

For example:

Dockerfile:

```
FROM openjdk:19-jdk
WORKDIR /app
EXPOSE 8080
COPY target/PFMS-1.0-SNAPSHOT.jar /app/app.jar
ENTRYPOINT ["java","-jar","/app/app.jar"]
```

2. Building the Image: The Docker image was created using the Docker CLI by running the command 'docker build -t pfms .' from the root of the project directory, which is where the Dockerfile is kept. It was then given a tag for convenient access.

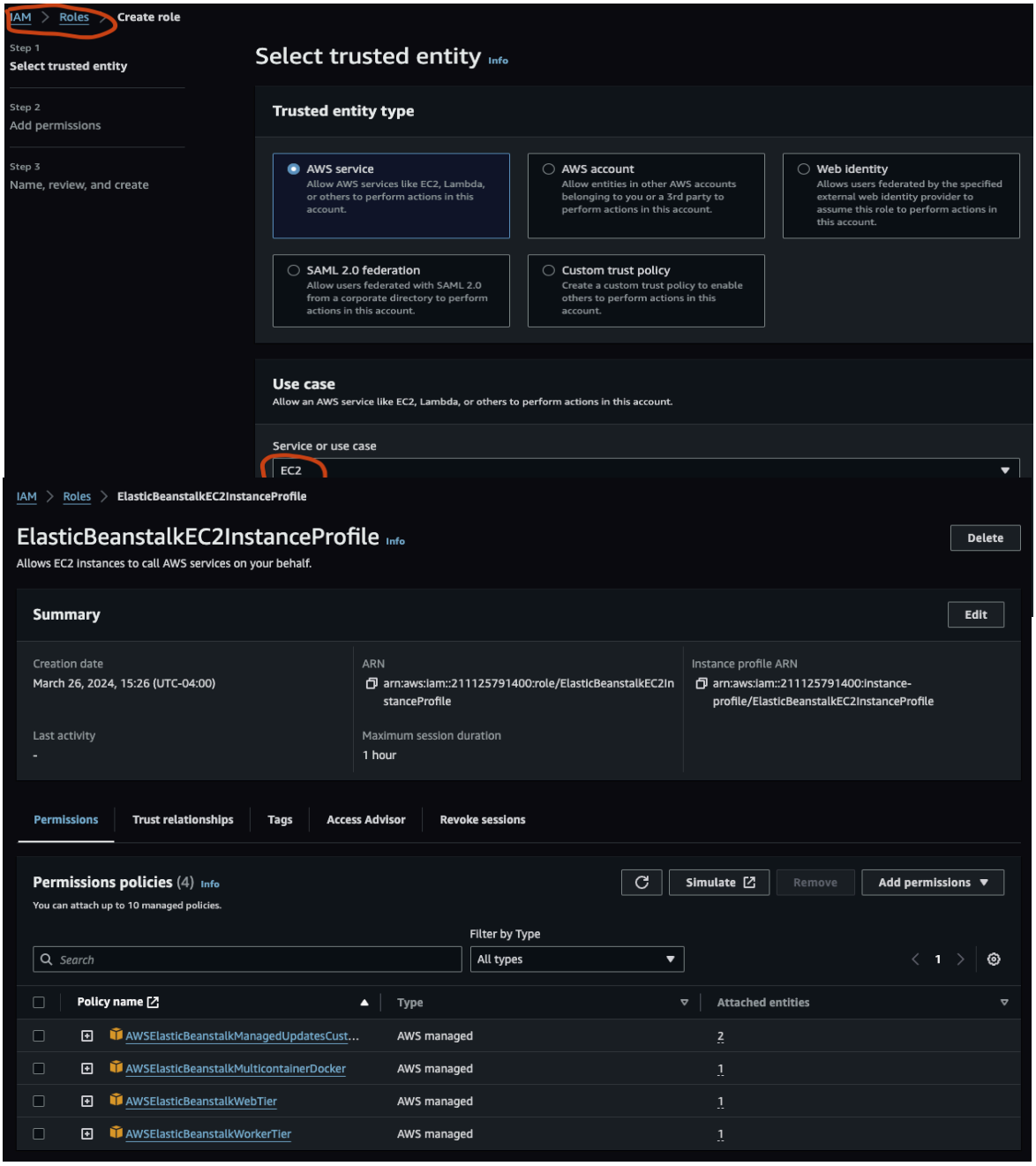
```
docker build -t pfms .
```

3. Local Testing: To make sure the application functioned as intended inside the container, the Docker image was tested locally prior to deployment to AWS. The command docker 'run -p 8080:8080 pfms' was used to do this.

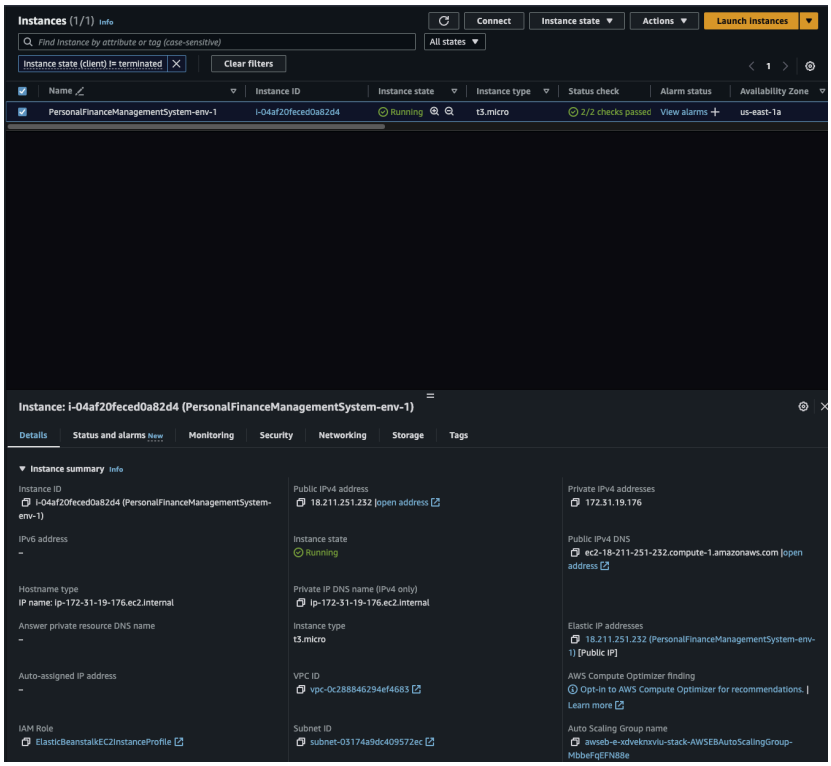
```
docker run -p 8080:8080 pfms
http://localhost:8080
```

4. Deployment Preparation: To enable the application to be deployed in a containerized fashion, an environment supporting Docker was set up on the AWS Elastic Beanstalk (web deploying) platform. First I would have to create an IAM (Identity and Access Management) role for an EC2 (cloud computing platforms) instance, then use that instance profile for Elastic Beanstalk.

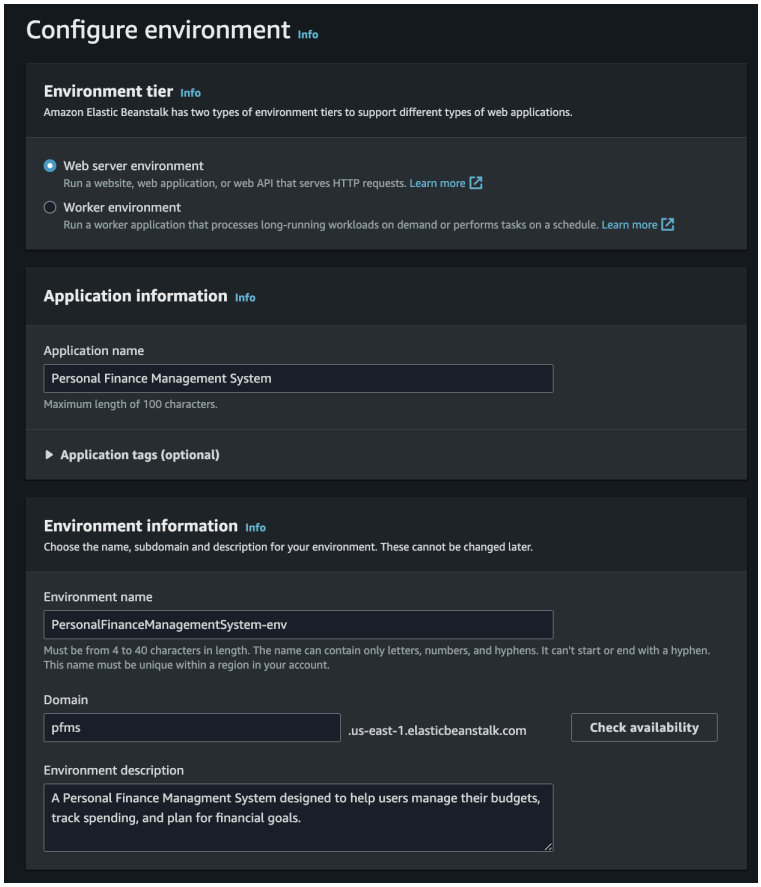
Creating an IAM (Identity and Access Management) role for EC2 instance that'll be used for the Elastic Beanstalk environment:

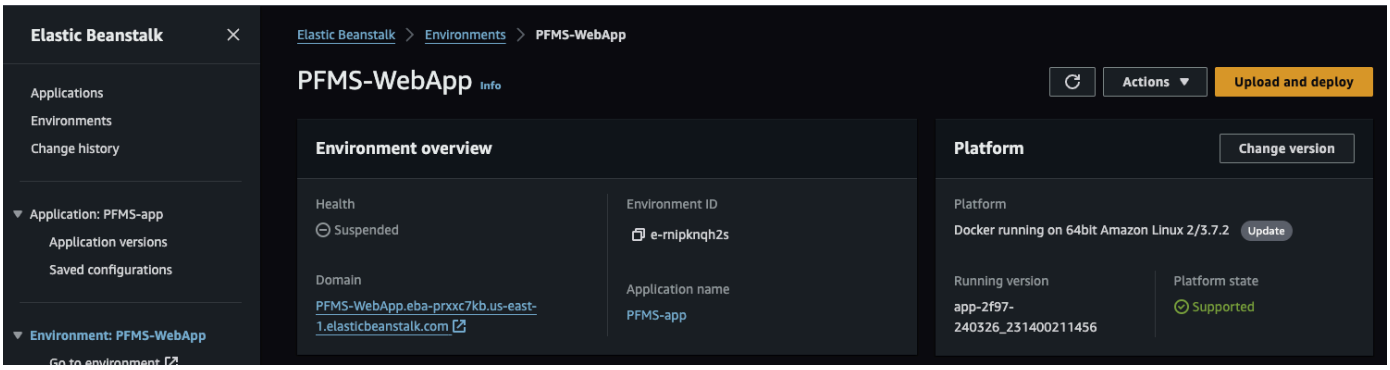


The EC2 instance for Elastic Beanstalk:



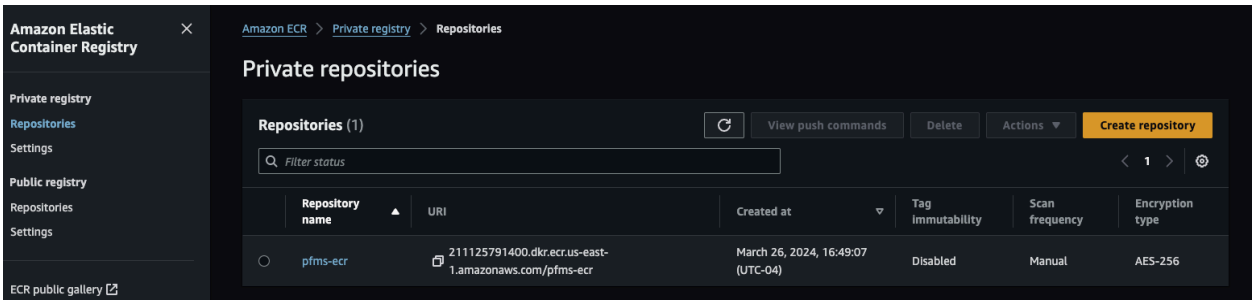
Setting up AWS Elastic Beanstalk environment:



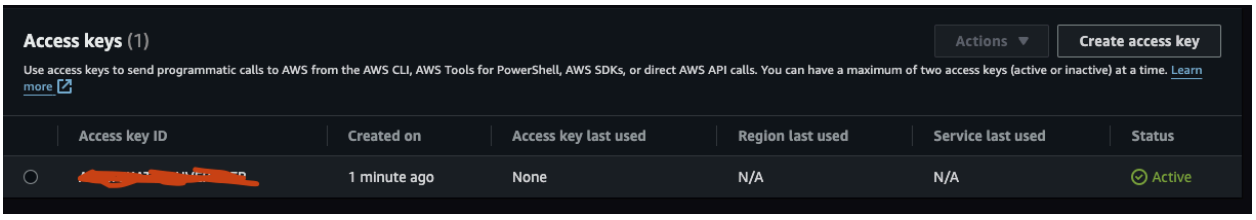


5. Image Deployment: After the Docker image was then deployed to the Elastic Beanstalk environment, I can use ECR (Amazon Elastic Container Registry) to store the Docker image, which will be referenced in a 'Dockerrun.aws.json' file.

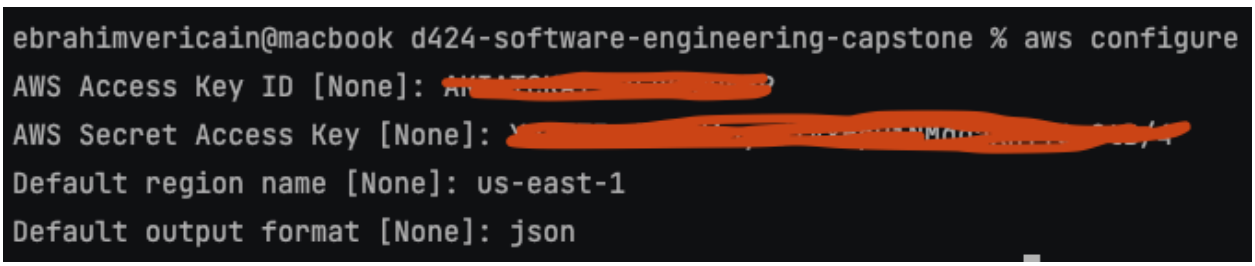
Creating ECR repository to contain the Docker image:



NOTE: To access AWS ECR through the terminal, I would need to enter the aws configuration credentials first. Before that, I would need to create an access key.



**Entering the aws configuration credentials:*



Authenticating:

```
ebrahimvericain@macbook d424-software-engineering-capstone % aws ecr get-login-password --region us-east-1 | docker lo
gin --username AWS --password-stdin 211125791400.dkr.ecr.us-east-1.amazonaws.com

Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/a
ccess-tokens/
ebrahimvericain@macbook d424-software-engineering-capstone %
```

Tagging Docker image:

```
ebrahimvericain@macbook d424-software-engineering-capstone % docker tag pfms:latest 211125791400.dkr.ecr.us-east-1.ama
zonaws.com/pfms-ecr:latest
```

```
ebrahimvericain@macbook d424-software-engineering-capstone % docker images
REPOSITORY                                          TAG       IMAGE ID       CREATED        SIZE
pfms                                                latest    3d71d4297483   26 hours ago   539MB
211125791400.dkr.ecr.us-east-1.amazonaws.com/pfms-ecr   latest    3d71d4297483   26 hours ago   539MB
```

Pushing the Docker image to the ECR repo:

```
ebrahimvericain@macbook d424-software-engineering-capstone % docker push 211125791400.dkr.ecr.us-east-1.amazonaws.com/
pfms-ecr:latest

The push refers to repository [211125791400.dkr.ecr.us-east-1.amazonaws.com/pfms-ecr]
cbf5a360b63e: Pushed
d11cf6312476: Pushed
9037429cb38f: Pushed
2d782770310f: Pushed
abfe99b76a78: Pushed
latest: digest: sha256:1877b02db395d9bb3df225578848c9b813d41a567e74f4d434ee7c6bf062f03d size: 1372
```

Creating the Dockerrun.aws.json file:

db.sql | {} Dockerrun.aws.json

Users > ebrahimvericain > IdeaProjects > personal-finance-management-system > {} Dockerrun.aws.json > ...

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "211125791400.dkr.ecr.us-east-1.amazonaws.com/pfms-ecr:latest",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "8080"
    }
  ]
}
```

Deploy on AWS Elastic Beanstalk:

```
ebrahimvericain@macbook d424-software-engineering-capstone % eb init -r us-east-1
```

```
ebrahimvericain@macbook d424-software-engineering-capstone % eb create PFMS-WebApp
Creating application version archive "app-2f97-240326_230735114838".
Uploading: [##################################################] 100% Done...
Environment details for: PFMS-WebApp
  Application name: PFMS-app
  Region: us-east-1
  Deployed Version: app-2f97-240326_230735114838
  Environment ID: e-rnipknqh2s
  Platform: arn:aws:elasticbeanstalk:us-east-1::platform/Docker running on 64bit Amazon Linux 2/3.7.2
  Tier: WebServer-Standard-1.0
  CNAME: UNKNOWN
  Updated: 2024-03-27 03:07:49.297000+00:00
Printing Status:
2024-03-27 03:07:47    INFO    createEnvironment is starting.
2024-03-27 03:07:49    INFO    Using elasticbeanstalk-us-east-1-211125791400 as Amazon S3 storage bucket for environment data.
2024-03-27 03:08:11    INFO    Created security group named: sg-03b7eb024aaec907e
2024-03-27 03:08:11    INFO    Created security group named: awseb-e-rnipknqh2s-stack-AWSEBSecurityGroup-quauWTwHjSw3
```

```
ebrahimvericain@macbook d424-software-engineering-capstone % eb status
Environment details for: PFMS-WebApp
  Application name: PFMS-app
  Region: us-east-1
  Deployed Version: app-2f97-240326_230735114838
  Environment ID: e-rnipknqh2s
  Platform: arn:aws:elasticbeanstalk:us-east-1::platform/Docker running on 64bit Amazon Linux 2/3.7.2
  Tier: WebServer-Standard-1.0
  CNAME: PFMS-WebApp.eba-prxxc7kb.us-east-1.elasticbeanstalk.com
  Updated: 2024-03-27 03:12:21.464000+00:00
  Status: Ready
  Health: Yellow
ebrahimvericain@macbook d424-software-engineering-capstone % eb deploy
Creating application version archive "app-2f97-240326_231400211456".
Uploading: [##################################################] 100% Done...
2024-03-27 03:14:10    INFO    Environment update is starting.
2024-03-27 03:14:15    INFO    Deploying new version to instance(s).
2024-03-27 03:14:31    INFO    Instance deployment completed successfully.
2024-03-27 03:14:37    INFO    New application version was deployed to running EC2 instances.
2024-03-27 03:14:37    INFO    Environment update completed successfully.
```

Verifying is the web app is successfully hosted: